IMPLEMENTASI METODE NAÏVE BAYES UNTUK PENGKATEGORIAN CALON PESERTA KKN STUDI KASUS : CALON PESERTA KKN UNIVERSITAS KRISTEN DUTA WACANA

Herlika Sigalingging
Wimmie Handiwidjojo, Yetli Oslan

Abstraksi

KKN merupakan salah satu wujud tindak nyata bentuk pengabdian kepada masyarakat pedesaan yang dilakukan oleh beragam latar belakang mahasiswa. Di UKDW terdapat beberapa jenis KKN yang dapat diikuti oleh tiap mahasiswa, diantaranya : KKN Reguler, KKN YKFS (Yogyakarta Kampung Field School), dan KKN Service-Learning. Mahasiswa yang memiliki kemampuan dalam bidang akademik maupun softskills akan dapat menunjang proses KKN berjalan dengan lancar, sehingga tujuan yang ditetapkan dapat tercapai. Masalah muncul ketika proses penerimaan dan pengkategorian calon peserta KKN dituntut untuk memberikan informasi hasil pengkategorian calon peserta KKN yang sesuai dengan jenis KKN dengan cepat kepada calon peserta KKN dan proses administrasi kelengkapan KKN yang dapat memudahkan dari sisi calon peserta KKN maupun pengambil keputusan.

Untuk memudahkan proses administrasi dan pengkategorian calon peserta KKN maka dibutuhkan suatu sistem yang memudahkan pengguna dalam melakukan pendaftaran KKN dan memilih calon peserta KKN yang layak mengikuti KKN sesuai dengan jenis KKNnya menggunakan metode Naïve Bayes. Metode Naïve Bayes berguna untuk melakukan pengkategorian calon peserta KKN yang sesuai dengan kriteria IPK, total SKS, jenis kelamin, dan kesehatan calon peserta KKN dan kemudian menghitung nilai-nilai tersebut dengan rumus tertentu untuk dijadikan nilai pada jenis-jenis KKN agar calon peserta KKN dapat melihat hasil rekomendasi jenis KKN dan menentukan pilihannya.

Penelitian ini mampu menghasilkan program yang akan membantu mencari calon peserta KKN yang memenuhi syarat administrasi dan merekomendasikan calon peserta KKN sesuai dengan jenis KKNnya menggunakan metode Naïve Bayes. Program ini dibuat berbasis web agar mempermudah dalam proses pendaftaran KKN secara online serta memberikan rekomendasi jenis KKN yang sesuai kriteria dari pengambil keputusan. Program membantu pihak pengambil keputusan dalam mengategorikan calon peserta KKN yang sesuai dengan jenis KKN berdasarkan syarat KKN yang ada dan bobot yang dimasukkan..

Kata Kunci : *Pengkategorian, Naive Bayes, Kuliah Kerja Nyata*

1 Pendahuluan

KKN merupakan salah satu wujud tindak nyata bentuk pengabdian kepada masyarakat pedesaan yang dilakukan oleh beragam latar belakang mahasiswa. Mahasiswa yang memiliki kemampuan dalam bidang akademik maupun *softskills* akan dapat menunjang proses KKN berjalan dengan lancar, sehingga tujuan yang ditetapkan dapat tercapai.

Untuk memilih mahasiswa sesuai dengan jenis KKN tersebut dilakukan dengan cara pengkategorian calon peserta KKN, supaya para peserta KKN yang terpilih nantinya memiliki kecocokan dengan kemampuan akademik maupun *softskills* sesuai dengan jenis KKN yang tersedia. Masalah timbul ketika pihak LPPM dalam proses pemilihan dan pengkategorian yang dilakukan masih menggunakan cara manual, sehingga memerlukan waktu yang lama dalam proses pengolahannya. Contohnya, penilaian yang bersifat subyektif, kesulitan dalam menemukan data, karena adanya penumpukan dokumen peserta, serta keterlambatan pengambilan keputusan.

Karena terdapat kesulitan dalam memilih calon peserta yang layak mengikuti KKN dan mengategorikan calon peserta sesuai dengan jenis KKN berdasarkan kriteria dari pengambil keputusan, maka dibutuhkan suatu sistem aplikasi yang dapat membantu pihak LPPM untuk memilih

calon peserta yang layak mengikuti KKN dan mengategorikan calon peserta sesuai dengan jenis KKN berdasarkan kriteria dari LPPM dalam waktu yang singkat. Diharapkan ada sebuah sistem berbasis web yang dapat mengelompokkan kecocokan antara jenis KKN dengan mahasiswa yang mendaftar sebagai peserta KKN.

2 Tinjauan Pustaka

2.1 Desain Web

Web Design adalah istilah yang sering digunakan untuk menggambarkan bagaimana tampilan isi suatu *website* atau situs. Dalam tahap pembuatan desain web hal yang paling penting diperhatikan selain dari segi kegunaan, fungsi, isi, ataupun pengalaman pengguna (*user's experience*), bagaimana informasi yang terdapat dalam web tersebut disampaikan kepada pengguna dalam hal ini web harus dapat diorganisasikan dan di"*link*"an dengan baik, sehingga tidak menyulitkan bagi pengguna yang menggunakan web tersebut (Powell, 1998:2).

2.2 Metode Naive Bayes

Metode Naive Bayes adalah metode yang dapat digunakan untuk melakukan klasifikasi dokumen secara statistik. Metode ini dapat memprediksi probabilitas keanggotaan suatu kelas dari suatu data. Teorema Bayes menurut Han & Kamber, (2006:42) adalah sebagai berikut.

$$P(C_i|X) = \frac{P(X|Ci).P(Ci)}{P(X)} \quad \text{--- (1)}$$

Dimana :

P(X) = probabilitas konstan untuk semua kelas sehingga hanya bentuk P(X|Ci).P(Ci) yang dimaksimumkan.

P(Ci) = probabilitas yang dihitung melalui training data.

P(X|Ci) = probabilitas yang nantinya akan dihitung dengan rumus.

HMAP (*Hypothesis Maximum Appropri Probability*) menyatakan hipotesa yang diambil berdasarkan nilai probabilitas dari kondisi prior yang diketahui. HMAP dapat dihitung dengan menggunakan perumusan sebagai berikut :

$$\text{HMAP} = \text{argmax } P(C_i|X) = \text{argmax } P(X|C_i).P(C_i) \quad \text{--- (2)}$$

Metode Bayes hanya dapat digunakan untuk menyelesaikan permasalahan klasifikasi dengan *supervised learning* dan data-data kategorikal, maka di dalam penggunaannya, metode Bayes memerlukan pengetahuan awal di dalam mengambil keputusan. Tingkat keberhasilan dari metode ini sangat tergantung pada training data yang diberikan.

2.3 ASP.Net

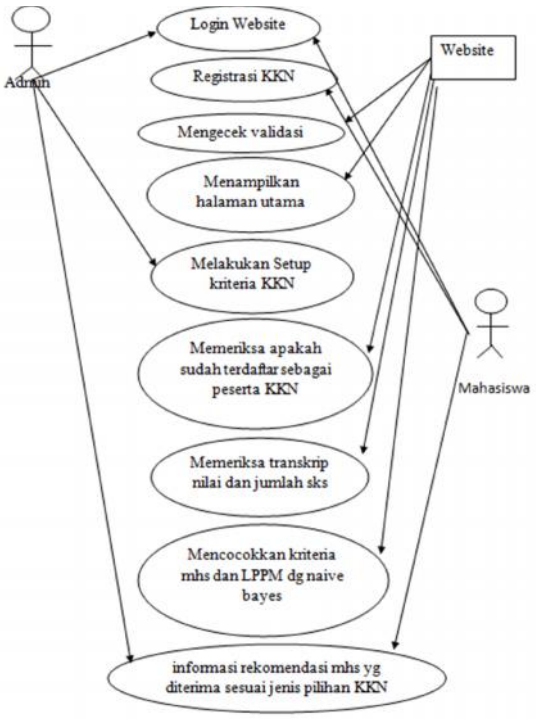
ASP.NET Control merupakan jantung dari ASP.NET *Framework*. ASP.NET Control adalah class .NET yang dieksekusi di *server*, yang kemudian hasilnya ditampilkan ke *browser* (Kurniawan, 2010 : 25). ASP.NET *Framework* memiliki sekitar 70 *control* yang digunakan untuk menampilkan *list record database* sampai menampilkan *banner rotation* secara random.

3 Rancangan Sistem dan Database

3.1 Rancangan Sistem

Proses perancangan ini bertujuan agar dalam pembuatan sistem informasi pengkategorian calon peserta KKN ini dapat sesuai dengan konsep yang sudah direncanakan. Bahasa pemrograman yang

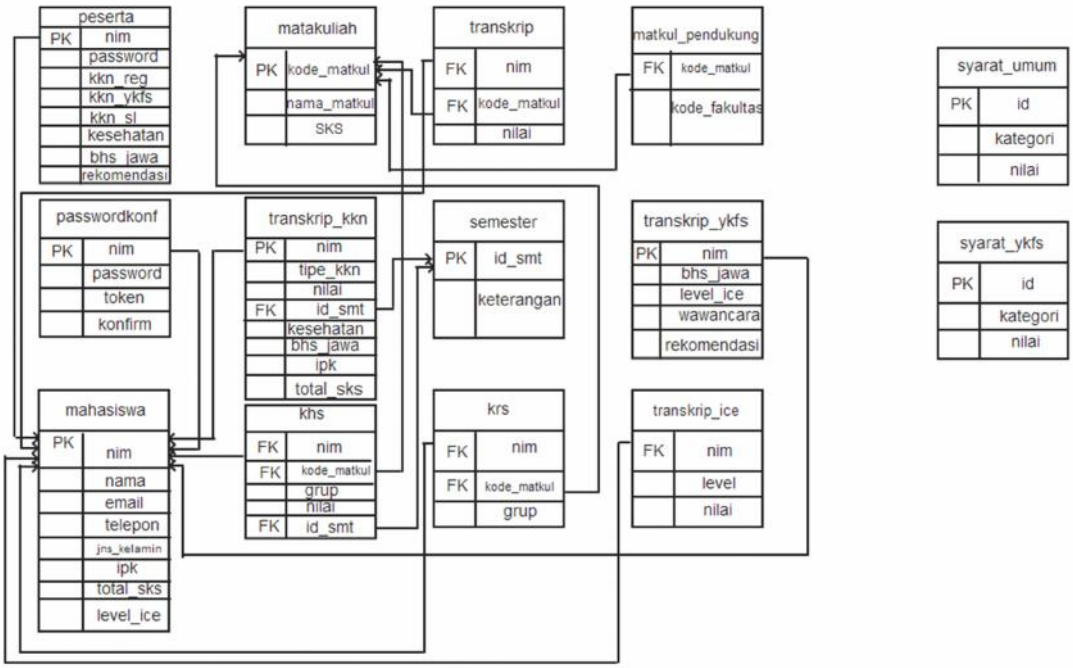
digunakan adalah C# dengan tools Microsoft Visual Studio 2010 dan tempat penyimpanan database menggunakan SQL Server Management Studio 2008 R2. Pengguna yang terlibat dalam sistem informasi ini adalah pihak LPPM sebagai pengguna utama program dan yang melakukan pengaturan data KKN serta mahasiswa UKDW yang menjadi calon peserta KKN sebagai pemberi masukan informasi yang dibutuhkan sistem. Use case diagram untuk sistem informasi pengkategorian calon peserta KKN dapat dilihat pada Gambar 1



Gambar 1: Use Case Diagram Pengkategorian Calon Peserta KKN

3.2 Basis Data

Gambar 2 menunjukkan database yang digunakan dalam program. Tabel yang terlihat antara lain tabel `mahasiswa`, `khs`, `krs`, `transkrip`, `matakuliah`, `semester`, `transkrip_ice` merupakan tabel-tabel referensi yang diambil dari sistem akademik yang ada dikampus UKDW yang menyimpan data tentang akademik mahasiswa, sedangkan tabel `peserta`, `passwordkonf`, `matkul_pendukung`, `transkrip_kkn`, `transkrip_ykfs`, `syarat_umum`, `syarat_ykfs` merupakan tabel yang dibuat untuk sistem informasi pengkategorian calon peserta KKN UKDW yang menyimpan tentang hasil rekomendasi jenis KKN yang telah dihitung menggunakan metode Naïve Bayes. tabel `peserta` menyimpan data mahasiswa yang mendaftarkan diri sebagai calon peserta KKN di web. Tabel `passwordkonf` menyimpan data password peserta KKN agar dapat login ke sistem. Tabel `matkul_pendukung` menyimpan data matakuliah pendukung mahasiswa. Tabel `transkrip_kkn` menyimpan data mahasiswa yang telah mengikuti KKN berdasarkan jenis KKNnya. Tabel `transkrip_ykfs` menyimpan data rekomendasi calon peserta yang layak diterima di KKN YKFS. Tabel `syarat_umum` dan `syarat_ykfs` menyimpan data persyaratan KKN.

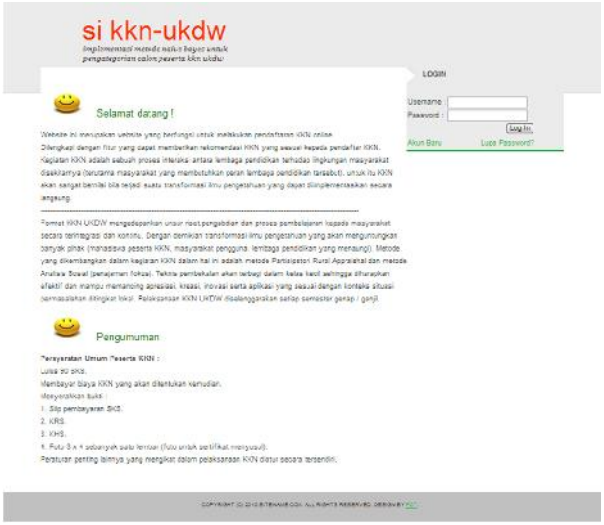


Gambar 2: Rancangan Basis Data

4 Implementasi Sistem dan Analisis Sistem

4.1 Implementasi Sistem

Pengguna yang ingin login kedalam sistem ini terlebih dahulu harus mendaftarkan diri di web ini kecuali admin. Halaman default awal ketika pengguna mengakses web ini dapat dilihat pada Gambar 3. Pada halaman default terdapat link dibagian sidebar kanan untuk melakukan registrasi akun pengguna. Ketika pengguna mengklik link "Akun Baru", maka sistem akan mengarahkan pengguna ke halaman registrasi secara otomatis. Pengguna yang ingin mendaftarkan diri di web ini terlebih dahulu memastikan bahwa data pribadinya harus sudah terdaftar sebagai mahasiswa yang datanya disimpan pada tabel `mahasiswa`.

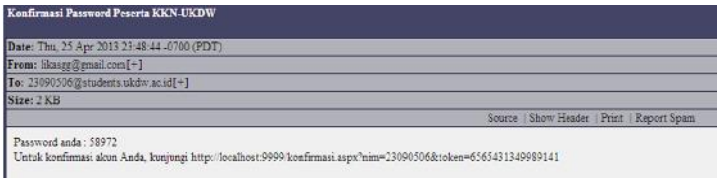


Gambar 3: Diagram alir sistem

Ketika link “Akun Baru” diklik, maka akan tampil halaman registrasi seperti terlihat pada Gambar 4. Terdapat beberapa input yang harus dimasukkan oleh pengguna. Pengguna harus memasukkan data nim, dan kemudian sistem akan melakukan pencarian pada tabel `mahasiswa` yang sesuai dengan input nim yang dimasukkan pengguna. Setelah itu, pengguna harus memasukkan data IPK terakhir pada saat registrasi. Pada saat tombol “Daftar” ditekan, sistem akan menyaring calon peserta yang memenuhi persyaratan administrasi KKN yang disimpan dalam tabel `syarat_umum` yaitu sudah memenuhi syarat terdaftar sebagai peserta matakuliah KKN di fakultas masing-masing yang diambil dari tabel `KRS`. Setelah itu, sudah menempuh 90sks tanpa nilai E yang diambil dari tabel `transkrip`, dan IPK 2.00 pada saat registrasi yang diambil dari tabel `mahasiswa`.

Gambar 4: Halaman Registrasi

Selanjutnya, apabila telah berhasil melakukan registrasi maka sistem akan memberikan pesan informasi kepada pengguna untuk memastikan bahwa syarat surat kesehatan telah terpenuhi dengan bukti surat kesehatan yang telah diberikan oleh pihak LPPM. Setelah itu, admin akan langsung segera melakukan pemeriksaan data surat kesehatan pengguna, apabila sudah terpenuhi, maka sistem akan mengirimkan password kepada pengguna untuk selanjutnya dilakukan konfirmasi melalui pesan link yang terdapat dalam email pengguna, agar pengguna dapat login ke sistem. Password akan dikirimkan ke email students pengguna seperti terlihat pada Gambar 5. Data password dan identitas pengguna disimpan dalam tabel `passwordkonf`.



Gambar 5: Konfirmasi Password Peserta KKN

Proses peyajian informasi selanjutnya kepada pengguna adalah kembali ke halaman default sistem untuk dapat melakukan login. Ketika login berhasil dilakukan, terdapat halaman yang berisi informasi tentang pilihan jenis KKN yang diinginkan serta pertanyaannya lainnya yang nantinya pada saat diproses maka analisis Naïve Bayes akan diterapkan. Untuk memilih jenis pilihan KKN yang diinginkan maka pengguna dapat mencentang jenis pilihan KKN yang disediakan oleh sistem seperti terlihat pada Gambar 6. Nilai yang berasal dari perhitungan Naïve Bayes ini yang akan dipakai oleh pengguna untuk mempertimbangkan pilihan jenis KKN yang akan dipilih.

Gambar 6 : Halaman Pilih Jenis KKN

Tombol dengan label “Proses” pada Gambar 6 mengeksekusi perhitungan Naïve Bayes dengan rumus yang sudah dijelaskan pada bagian 2.2. Pada perhitunga ini, sistem akan menggunakan kriteria IPK,jumlah sks, jenis kelamin, serta pilihan yang telah dipilih oleh pengguna pada Gambar 6. Hasil perhitungan Naïve Bayes tersebut adalah muncul pesan informasi yang berisi hasil rekomendasi pilihan jenis KKN dan prosentase masing-masing jenis KKN berdasarkan perhitungan Naïve Bayes. Hasil rekomendasi yang memiliki tingkat prosentase paling tinggi adalah hasil rekomendasi yang paling sesuai dengan kriteria dari pengguna dan kriteria dari pengambil keputusan dan hasil probabilitas harus > 0.

Berikut ini merupakan *source code* yang digunakan dalam proses Naïve Bayes. Langkah pertama dalam proses Naïve Bayes adalah mendeklarasikan kriteria yang akan digunakan dalam proses perhitungan Naïve Bayes. berikut merupakan potongan *source code*nya.

```
publicstaticKriteria[]  kriteria  =  {  newKriteria("transkrip_kkn","ipk",JenisKriteria.range,
newList<Range>
{
newRange(2,(decimal)2.5),
newRange((decimal)2.51,3),
newRange((decimal)3.01,4)}),
newKriteria("transkrip_kkn","total_sks",JenisKriteria.range, newList<Range>
{

newRange(90,100),
newRange(101,110),
newRange(120,Decimal.MaxValue)}),
newKriteria("mahasiswa","jns_kelamin",JenisKriteria.fix)
};
```

Penjelasan *source code* diatas mendeklarasikan bagian yang pertama adalah membuat range untuk IPK, dan yang kedua mendeklarasian range total sks. Setelah itu, membuat variabel yang menyimpan jumlah data yang mengikut KKN di tiap jenis KKN. Berikut merupakan potongan *source code*nya.

```
int jml_reg = (int)conn.get("select count(*) from transkrip_kkn where tipe_kkn=1");
int jml_ykfs = (int)conn.get("select count(*) from transkrip_kkn where tipe_kkn=2");
int jml_sl = (int)conn.get("select count(*) from transkrip_kkn where tipe_kkn=3");
```

Selanjutnya memasukkan data kriteria peserta yang mendaftar KKN. Berikut potongan *source code*nya.

```
DataTable dta = conn.getdata("select nim from peserta where kesehatan=0 and nim!='admin'");
foreach (DataRow dr in dta.Rows)
            {
                        lblPesan.Text   +=   dr["nim"]+"   :   Cek   Kesehatan   peserta   terlebih
                        dahulu!<br>";
            }
```

Berikut merupakan penjelasan dari potongan *source code* diatas. Sistem akan mencocokan data yang mendaftar KKN dengan data yang berada di tabel peserta. Jika, peserta telah melakukan pendaftaran dan mendapatkan konfirmasi bahwa surat kesehatan telah diberikan ke LPPM. Maka, proses perhitungan akan dilakukan, tetapi jika belum menyerahkan surat kesehatan, maka sistem akan memunculkan pesan informasi untuk memeriksa kelengkapan surat kesehatan terlebih dahulu.

Selanjutnya adalah proses Naïve Bayes dilakukan dengan menghitung probabilitas masing-masing kelas untuk setiap alternatif jenis KKN. Pada proses ini tabel yang digunakan adalah tabel transkrip_kkn yang berisi data mahasiswa yang telah mengikuti KKN berdasarkan jenis KKN, tabel peserta, tabel mahasiswa. Karena datanya menggunakan banyak tabel sehingga proses ini menggunakan perintah *select count from* (*table name*) *left outer join* (*table name*). Untuk menghitung probabilitas kelas masing-masing berdasarkan jenis KKNnya, berikut potongan *source code*nya :

```
foreach (Kriteria k in kriteria)
   {
if (k.jenis == JenisKriteria.fix)
          {
                int jumlah = (int)conn.get("select count (*) from transkrip_kkn left outer join
                mahasiswa on mahasiswa.nim=transkrip_kkn.nim where tipe_kkn=1 and " + k.tabel +
                "." + k.nama + "=@0", dr[k.nama]);
                sreg += String.Format("{0:f} * ", (double)jumlah / jml_reg);
reguler *= (decimal)jumlah / jml_reg;
```

```
                lblPesan.Text += String.Format("P('{0}={4}'|'Reguler') = {1}/{2} = {3:f}<br>",
                k.nama, jumlah, jml_reg, (double)jumlah / jml_reg,dr[k.nama]);

                jumlah = (int)conn.get("select count (*) from transkrip_kkn left outer join
                mahasiswa on mahasiswa.nim=transkrip_kkn.nim where tipe_kkn=2 and " + k.tabel +
                "." + k.nama + "=@0", dr[k.nama]);
                sykfs += String.Format("{0:f} * ", (double)jumlah / jml_ykfs);
        ykfs *= (decimal)jumlah / jml_ykfs;
                lblPesan.Text  +=  String.Format("P('{0}={4}'|'Ykfs')  =  {1}/{2}  =  {3:f}<br>",
                k.nama, jumlah, jml_ykfs, (double)jumlah / jml_ykfs, dr[k.nama]);

                jumlah = (int)conn.get("select count (*) from transkrip_kkn left outer join
                mahasiswa on mahasiswa.nim=transkrip_kkn.nim where tipe_kkn=3 and " + k.tabel +
                "." + k.nama + "=@0", dr[k.nama]);
                ssl += String.Format("{0:f} * ", (double)jumlah / jml_sl);
sl *= (decimal)jumlah / jml_sl;
                lblPesan.Text  +=  String.Format("P('{0}={4}'|'Sl')  =  {1}/{2}  =  {3:f}<br>",
                k.nama, jumlah, jml_sl, (double)jumlah / jml_sl, dr[k.nama]);
                     }
elseif (k.jenis == JenisKriteria.range)
                     {
                              decimal nilai = Convert.ToDecimal(dr[k.nama]);
Range r = k.getRange(nilai);
                              int jumlah = (int)conn.get("select count (*) from transkrip_kkn
                              left outer join mahasiswa on mahasiswa.nim=transkrip_kkn.nim
                              where tipe_kkn=1 and " + k.tabel + "." + k.nama + " BETWEEN @0
                              AND @1", r.min,r.max);
                        sreg += String.Format("{0:f} * ", (double)jumlah / jml_reg);
                      reguler *= (decimal)jumlah / jml_reg;
                              lblPesan.Text  +=  String.Format("P('{0}({4:g}-{5:g})'|'Reguler')
                              = {1}/{2} = {3:f}<br>", k.nama, jumlah, jml_reg, (double)jumlah
                              / jml_reg,r.min,r.max);

                               jumlah = (int)conn.get("select count (*) from transkrip_kkn
                              left outer join mahasiswa on mahasiswa.nim=transkrip_kkn.nim
                              where tipe_kkn=2 and " + k.tabel + "." + k.nama + " BETWEEN @0
                              AND @1", r.min, r.max);
sykfs += String.Format("{0:f} * ", (double)jumlah / jml_ykfs);
                      ykfs *= (decimal)jumlah / jml_ykfs;
                              lblPesan.Text   +=   String.Format("P('{0}({4}-{5})'|'Ykfs')   =
                              {1}/{2} = {3:f}<br>", k.nama, jumlah, jml_ykfs, (double)jumlah /
                              jml_ykfs,r.min, r.max);

                               jumlah = (int)conn.get("select count (*) from transkrip_kkn
                              left outer join mahasiswa on mahasiswa.nim=transkrip_kkn.nim
                              where tipe_kkn=3 and " + k.tabel + "." + k.nama + " BETWEEN @0
                              AND @1", r.min, r.max);
 ssl += String.Format("{0:f} * ", (double)jumlah / jml_sl);
                         sl *= (decimal)jumlah / jml_sl;
                              lblPesan.Text += String.Format("P('{0}({4}-{5})'|'Sl') = {1}/{2}
                              = {3:f}<br>", k.nama, jumlah, jml_sl, (double)jumlah / jml_sl,
                              r.min, r.max);

                     }
                 }
```

Potongan *source code* diatas menjelaskan perhitungan *Conditional Probabilities P(Xt|Ci)* untuk setiap nilai atribut dari sampel data X dengan menggunakan *training data set* dan menghitung probabilitas *P(X|Cn)* sampel data X terhadap masing-masing kelas yang ada.

Setelah masing-masing kelas telah mendapatkan probabilitas, maka proses selanjutnya akan dilakukan proses perhitungan HMAP (*Hyphothesis Maximum Appropri Probability*) dan memberikan output nilai jenis KKN dengan probabilitas tertinggi. Sehingga dapat diketahui bahwa data peserta KKN yang mendaftar KKN dapat direkomendasikan ke jenis KKN yang sesuai dengan kriteria. Berikut merupakan potongan *source code*nya.

```
lblPesan.Text += "<br>";
decimal hmapReg = reguler * jml_reg / jml;
decimal hmapYkfs = ykfs * jml_ykfs / jml;
decimal hmapSl = sl * jml_sl / jml;

lblPesan.Text += String.Format("P(X|'Reguler') = {0} = {1:f}<br>", sreg, reguler);
lblPesan.Text += String.Format("P(X|'Ykfs') = {0} = {1:f}<br>", sykfs, ykfs);
lblPesan.Text += String.Format("P(X|'Sl') = {0} = {1:f}<br><br>", ssl, sl);
```

```
lblPesan.Text += String.Format("P(X|'Reguler')*P('Reguler') {0:f}*{1:f} ={2:f}<br>", reguler, (double)jml_reg / jml, hmapReg);
lblPesan.Text   +=   String.Format("P(X|'Ykfs')*P('Ykfs')   {0:f}*{1:f}   ={2:f}<br>",   ykfs, (double)jml_ykfs / jml, hmapYkfs);
lblPesan.Text += String.Format("P(X|'Sl')*P('Sl') {0:f}*{1:f} ={2:f}<br>", sl, (double)jml_sl / jml, hmapSl);

                lblPesan.Text += "<br>";

int jenis = 0;

if (hmapReg>hmapYkfs && hmapReg>hmapSl)
                {
        lblPesan.Text += "<br>";
                                lblPesan.Text += dr["nim"] + " Anda masuk ke dalam kategori KKN Reguler";
                    jenis = 1;
                }
elseif(hmapYkfs>hmapReg && hmapYkfs>hmapSl)
                {
                                lblPesan.Text += dr["nim"] + "Anda masuk ke dalam kategori KKN YKFS";
                    jenis = 2;
                }
elseif(hmapSl>hmapReg && hmapSl>hmapYkfs)
                {
                                lblPesan.Text += dr["nim"] + "Anda masuk ke dalam kategori KKN Service Learning";
                    jenis = 3;
                }
                lblPesan.Text += "<br><br>";
                                conn.executenonquery("update peserta set rekomendasi =@0 where nim=@1", jenis, dr["nim"]);
            }

            GridView1.Columns[6].Visible = true;
```

Penjelasan *source code* diatas, jika hasil perhitungan menghasilkan probabilitas regular lebih besar daripada YKFS dan Service Learning, maka sistem akan memunculkan pesan informasi bahwa peserta yang mendaftar direkomendasiikan dalam kategori KKN Reguler. Begitupula dengan dua jenis KKN yang lain, apabila hasil yang didapat lebih tinggi maka sistem akan membandingkan dengan hasil probabilitas KKN yang lain, kemudian akan memunculkan pesann informasi bahwa peserta direkomendasikan ke dalam kategori dengan hasil probilitas KKN tertinggi.

Gambar 7: Hasil Perhitungan Naive Bayes Peserta KKN

Nilai hasil rekomendasi dari perhitungan Naïve Bayes inilah yang digunakan baik dari calon peserta KKN maupun pengambil keputusan untuk dijadikan bahan pertimbangan dalam mengkategorikan jenis KKN mana berdasarkan kriteria dari calon peserta KKN dan syarat administrasi KKN. Hasil dari perhitungan Naïve Bayes ini akan disimpan dalam tabel `transkrip_kkn`.

4.2 Analisis Sistem

Pada bagian ini akan dijelaskan tentang tahapan analisis sistem yang dilakukan terhadap sampel data mahasiswa KKN. Berikut ini merupakan langkah-langkah dalam metode Naïve Bayes :

a. Menghitung *prior probabilities P(Ci) class* ke-i.
b. Menghitung *conditional probabilities* P(Xt|Ci) setiap atribut dari *class ke-i* terhadap sampel data X.
c. Menghitung *posterior probabilities P(X|Ci)* sampel data X terhadap *class ke-i.*

d. Menghitung *posterior probabilities* HMAP *P(Ci|X) class ke-i terhadap sampel data X.*

No	IPK	Total SKS	Jenis Kelamin	Jenis KKN
1.	3,44	149	Perempuan	Service Learning
2.	3,11	154	Perempuan	Reguler
3.	3,87	144	Laki – laki	Reguler
4.	3,95	143	Laki – laki	YKFS
5.	2,42	153	Perempuan	Reguler
6.	3,04	151	Laki – laki	Service Learning
7.	3,56	138	Perempuan	YKFS
8.	3,51	139	Laki – laki	Service Learning
9.	3,28	146	Laki – laki	YKFS
10.	3,70	131	Perempuan	YKFS
11.	3,46	134	Laki – laki	Service Learning
12.	3,39	132	Laki – laki	Reguler
13.	3,03	125	Perempuan	Reguler
14.	3,60	139	Perempuan	YKFS
15.	3,49	118	Laki – laki	Reguler
16.	3,45	131	Perempuan	Reguler
17.	2,98	140	Laki – laki	YKFS
18.	2,97	107	Perempuan	Reguler
19.	3,35	133	Laki – laki	Reguler
20.	3,21	112	Perempuan	Reguler
21.	3,89	121	Laki – laki	Reguler
22.	3,50	130	Laki – laki	Regular
23.	3,49	131	Laki – laki	Regular
24.	3,28	120	Laki – laki	Regular
25.	3,24	135	Laki – laki	Reguler

Gambar 8: Sampel Data Mahasiswa KKN

Pengguna menggunakan data sampel sebanyak 25 data yang memiliki kriteria peserta KKN dan jenis KKN yang berbeda. Terdapat sebuah contoh kasus yang digunakan untuk melakukan analisis terhadap sistem yang dibuat.

Studi kasus : Terdapat calon peserta yang ingin mendaftar KKN, memiliki IPK 3,20 , jumlah SKS 120, sehat jasmani dan rohani, berjenis kelamin Laki - laki. Jenis KKN apa yang sesuai dengan kriteria calon peserta KKN?

Berdasarkan contoh studi kasus diatas maka dapat diilustrasikan sebagai berikut:

X = {IPK="3,20", Jumlah SKS="120", Sehat="Ya", Jenis Kelamin="Laki - laki"}

Setelah itu, langkah-langkah metode Naïve Bayes yang sudah dipaparkan sebelumnya akan dijabarkan menggunakan data sampel.

a. Pertama, menghitung *prior probabilities* P(Ci) dari setiap jenis KKN :

P(Reguler) = 15/25 = 0.6

P(YKFS) = 6/25 = 0.24

P(SL) = 4/25 = 0.16

b. Kedua, menghitung *conditional probabilities* P(Xt|Ci) setiap atribut dari *class ke-i* terhadap sampel data X. Karena sampel calon peserta KKN memiliki kriteria IPK 3,20 sehingga calon peserta KKN masuk dalam range 3,01 - 4,00 seperti terlihat pada gambar 9 begitupula dengan menghitung probabilitas total sks dan jenis kelamin.

Kriteria	Nilai
IPK	2,00 – 2,50 2,51 – 3,00 3,01 – 4,00
Total SKS	90 – 100 101 – 110 111 – 120 >120
Jenis Kelamin	Laki – laki Perempuan

Gambar 9: Syarat KKN

Probabilitas P(IPK = "3.01 – 4.00" | Jenis KKN)

P(IPK = "3.01 – 4.00" | Reguler) = 13/15 = 0.86

P(IPK = "3.01 – 4.00" | YKFS) = 5/6 = 0.83

P(IPK = "3.01 – 4.00" | SL) = 4/4 = 1

Probabilitas P(Total_SKS = ">120" | Jenis KKN)

P(Total_SKS = ">120" | Reguler) = 12/15 = 0.8

P(Total_SKS = ">120" | YKFS) = 6/6 = 1

P(Total_SKS = “>120” | SL) = 4/4 = 1

Probabilitas P(Jenis_Kelamin = “Laki-laki” | Jenis KKN)

P(Jenis_Kelamin = “Laki-laki” | Reguler) = 9/15 = 0.6

P(Jenis_Kelamin = “Laki-laki” | YKFS) = 3/6 = 0.5

P(Jenis_Kelamin = “Laki-laki” | SL) = 3/4 = 0.75

c. Ketiga, menghitung posterior probabilities P(X|Ci) sampel data X terhadap class ke-i.

P(X|Reguler) = P(IPK=“3.01 – 4.00”|Reguler) *

P(Total_SKS =“>120”|Reguler) * P(Jenis_Kelamin=“Laki-laki”|Reguler)

P(X|Reguler) = 0.86 * 0.8 * 0.6 = 0.4128

P(X|YKFS) = P(IPK=“3.01 – 4.00”|YKFS) *P(Total_SKS=“>120”|YKFS) * P(Jenis_Kelamin=“Laki-laki”|YKFS)

P(X|YKFS) = 0.83 * 1 * 0.5= 0.415

P(X|SL) = P(IPK=“3.01 – 4.00”|SL) *P(Total_SKS=“>120”|SL) * P(Jenis_Kelamin=“Laki-laki”|SL)

P(X|SL) = 1* 1 * 0.75 = 0.75

d. Keempat, menghitung posterior probabilities P(Ci|X) class ke-i terhadap sampel data X.

P(Reguler|X) = P(X|Reguler) * P(Reguler) = 0.4128 * 0.6 = **0.24**

P(YKFS|X) = P(X|YKFS) * P(YKFS) = 0.415 * 0.24 = 0.09

P(SL|X) = P(X|SL) * P(SL) = 0.75 * 0.16 = 0.12

Dari perhitungan probabilitas masing-masing jenis KKN terhadap sampel data X, maka didapatkan jenis KKN dengan nilai probabilitas tertinggi yaitu KKN Reguler dengan probabilitas sebesar 0,24.

Tabel 1: Analisis Penerapan Algoritma Naive Bayes

IPK	Jmlsks	JK	Rekomendasi	KKN dipilih	Prioritas I	Prioritas II	Prioritas III
2,00	90	L	Reguler	YKFS, Reguler	Reguler	YKFS	SL
2,50	124	L	Reguler	SL	Reguler	YKFS	SL
3,46	123	P	Reguler	Reguler	Reguler	YKFS	SL
3,90	200	L	Reguler	Reguler	Reguler	YKFS	SL
2,89	145	P	Reguler	YKFS,Reguler,SL	Reguler	YKFS	SL
3,23	112	P	Regular	YKFS	Reguler	YKFS	SL
2,89	132	L	Regular	SL	Regular	YKFS	SL

Pengujian dilakukan menggunakan data dari mahasiswa yang telah mengikuti KKN sebanyak 560 data dan populasi mahasiswa yang mengikuti KKN Reguler sebanyak 391 mahasiswa, KKN YKFS sebanyak 90 mahasiswa dan KKN *Service Learning* sebanyak 79 mahasiswa.Dari hasil analisis yang dilakukan, hampir semua hasil analisis menghasilkan jenis KKN Reguler dengan probabilitas paling tinggi. Pengaruh dari adanya kriteria peserta dari segi ipk, total sks, dan jenis kelamin juga dapat mempengaruhi tingkat probabilitas dari jenis KKN.

Tabel 2: Analisis Penerapan Metode Naive Bayes

IPK	Jmlsks	JK	Rekomendasi	KKN dipilih	Prioritas I	Prioritas II	Prioritas III
3,90	200	L	YKFS	Reguler, YKFS, SL	YKFS	Reguler	SL
3,13	121	P	YKFS	YKFS	YKFS	SL	Reguler
3,87	145	L	YKFS	SL	YKFS	Reguler	SL
2,65	130	P	SL	Reguler	SL	Regular	YKFS
3,44	150	P	YKFS	Regular, SL	YKFS	SL	YKFS

Penulis melakukan uji coba terhadap data dengan menyeimbangkan jumlah data pada tiap jenis KKN berdasarkan jumlah data jenis KKN paling rendah yaitu Service Learning sebanyak 79 data.Dari hasis analisis yang dilakukan ternyata apabila terjadi pemerataan data di tiap jenis KKN akan menghasilkan rekomendasi hasil KKN yang bervariasi.

5 Kesimpulan

a. Sistem dapat mempermudah proses administrasi peserta KKN sehinga lebih efektif dan efisien terhadap waktu dan tenaga.
b. Metode Naïve Bayes kurang cocok diterapkan pada data penelitian yang dilakukan, karena tidak adanya pemerataan jumlah yang seimbang pada tiap jenis KKN, sehingga selalu menghasilkan probabilitas tertinggi yang mempunyai jumlah data terbanyak pada jenis KKN tersebut. Metode Naïve Bayes dapat diterapkan, jika adanya keseimbangan jumlah data pada tiap jenis-jenis KKN.
c. Sistem dapat memberikan informasi dan mempermudah pengguna dalam melakukan pendaftaran KKN.
d. Hasil pengategorian jenis KKN bergantung pada data, kriteria, dan bobot kriteria yang diatur oleh pengambil keputusan.

6 Daftar Pustaka

Han, Jiawei, Micheline Kamber. (2006). *Data Mining: Concepts and Techniques 2nd edition*. United States: Morgan Kaufmann.

Kurniawan, Erick. (2010). *Cepat Mahir ASP.NET 3.5 : Untuk Aplikasi Web Interaktif*. Yogyakarta: Andi Offset.

Powell, Thomas.A. (1998). *Web Site Engineering: Beyond Web Page Design*. United States: Prentice Hall PTR.